THIS OPINION WAS NOT WRITTEN FOR PUBLICATION

The opinion in support of the decision being entered today (1) was not written for publication in a law journal and (2) is not binding precedent of the Board.

Paper No. 28

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

Ex parte MICHAEL H. KELLEY

Appeal No. 95-1544
Application 08/003,602[1]

ON BRIEF

Before KRASS, JERRY SMITH and BARRETT, Administrative Patent Judges.

KRASS, Administrative Patent Judge.

DECISION ON APPEAL

[1]Application for patent filed January 13, 1993. According to appellant, the application is a continuation of Application 07/424,184 filed October 19, 1989, now abandoned.

This is a decision on appeal from the final rejection of claims 10 through 20, constituting all the claims in the application.

The invention is directed to a method of executing a special section of code, in a computer operating system, that must be completed without interruption by a page fault during execution. The nature of the method is apparent from a review of representative independent claim 10, reproduced as follows:

10. A method of executing a special section of code on an operating system of a computer system having an operating system, assigned storage, and memory references in secondary storage, wherein the special section of code performs a series of operations that must all be completed without interruption by a page fault during execution, thereby requiring all memory references to be in assigned storage at one time, comprising:

executing said special section of code on said operating system including requesting memory references and determining if a requested memory reference is not available in said assigned storage;

interrupting said executing the special section of code when it is determined that a requested memory reference is not available in said assigned storage;

prior to retrieving the unavailable memory reference from said secondary storage, undoing operations carried out by the special section of code so that substantially no executed fragments of the special section of code are in existence;

locating the unavailable memory reference in said secondary storage, retrieving the unavailable memory reference and writing the unavailable memory reference to said assigned storage; and

executing said special section of code on said operating system with the retrieved memory reference available in said assigned storage.

The examiner relies on the following reference:

Yamaguchi et al. (Yamaguchi) 5,003,458 Mar. 26, 1991 (filed Oct. 23, 1987)

Claims 10 through 20 stand rejected under 35 U.S.C. 102(e) as anticipated by Yamaguchi.

Reference is made to the briefs and answer for the respective positions of appellant and the examiner.

OPINION

At the outset, we note that this Board entered a decision in the parent application Serial No. 07/424,184 but the instant claims have been substantially amended vis á vis the claims which were the subject matter of the earlier appeal and the prior art reference here is different from the prior art reference relied on in the earlier appeal.

Anticipation, under 35 U.S.C. 102, requires that each element, or step, of the claim in issue be found, either expressly described or under principles of inherency, in a single prior art reference. Kalman v. Kimberly-Clark Corp., 713 F.2d 760, 771, 218 USPQ 781, 789 (Fed. Cir. 1983).

We have carefully considered the subject matter on appeal, the rejection advanced by the examiner and the evidence of anticipation relied on by the examiner as support for the rejection. Likewise, we have reviewed and taken into consideration appellant's arguments as set forth in the briefs along with the examiner's rationale in support of the rejection and arguments in rebuttal set forth in the answer.

It is our view, after consideration of the record before us, that the instant claims are not anticipated by the Yamaguchi reference.

Each of the independent claims requires that the "special section of code performs a series of operations that must be completed without interruption...requiring all memory references to be in assigned storage at one time" and

> prior to retrieving the unavailable memory reference from said secondary storage, undoing operations carried out by the special section of code so that substantially no executed fragments of the special section of code are in existence.[2]

---

[2]Claims 14, 16 and 18 do not recite "prior to retrieving the unavailable memory reference from said secondary storage" immediately preceding the "undoing operations" but these claims do require such a limitation. This is clear, in claims 14 and 16, by the recited first step of "modifying said page fault handling routine prior to executing said special section of code so that said page fault handling routine does not read a memory reference

The examiner identifies the "undoing" portion of the claim as being equivalent to that taught by Yamaguchi at column 5, lines 66 et seq. [answer, page 4]. In response to appellant's arguments, the examiner stated [answer, pages 6-7] that this is

> taught as the recovering of contents of the saved internal registers to return to a restart point. This teaches the limitations of "substantially no executed fragments of the special section of code are in existence". As Yamaguchi returns to a restart point for execution, the steps executed after the restart point are "erased". Yamaguchi has a "clean slate" to the point of the restart point in the program.

We have reviewed the portion of Yamaguchi cited by the examiner for the "undoing" limitation of the claims and while we agree that Yamaguchi discloses an instruction restart procedure for restarting an instruction after a page fault process, we find nothing in Yamaguchi which erases operations carried out by a special section of code, so that no executed fragments are in existence, before retrieving the unavailable memory reference from a secondary storage, thereby assuring that the entire series

---

from said secondary storage in response to a requested memory reference not being available in said assigned storage," and in claim 18, at step (c) "interrupting said executing said special section of code if it is determined in step (a) that a requested memory reference is not available in assigned storage" followed by the step of "undoing..."

of operations to be performed by the special section of code will be completed without interruption by a page fault because all memory references are assigned storage at one time.

While Yamaguchi discloses an instruction restart procedure for restarting an instruction after a page fault process, in this regard, Yamaguchi appears to disclose nothing more than the typical prior art procedure of first determining that a fault has occurred, then retrieving the unavailable memory reference from secondary storage and restarting the procedure at some point. This is in contrast to the instant claims which require that the operations carried out by the special section of code be undone, or erased, prior to the retrieval of the unavailable memory reference. Such a step permits the result recited in the claim preambles that the "special section of code performs a series of operations that must all be completed without interruption..." In Yamaguchi, the series of operations is interrupted and then memory reference retrieval is performed in order to restart the operations at some point.

One may interpret Yamaguchi to teach that a series of operations is completed prior to an interrupt and, as to those operations, there is certainly a completion of operation "without interruption." However, as appellant points out, at page 4 of

the reply brief, there is a difference between the instant claims, which require that the series of operations must be completed without interruption, and the teaching of Yamaguchi that a series of operations was completed without interruption. The former requires operation completion without interruption by the particular recited method of executing the special section of code while the latter may complete a series of operations without interruption but if an interruption occurs, Yamaguchi does not undo, or erase, all previous operations before retrieving the necessary unavailable memory reference.

Since we do not find, in Yamaguchi, all of the method steps of the instant claims, the examiner's decision rejecting claims 10 through 20 under 35 U.S.C. 102(e) as anticipated by Yamaguchi is reversed.

REVERSED

ERROL A. KRASS Administrative Patent Judge	)	
JERRY SMITH Administrative Patent Judge	)	BOARD OF PATENT APPEALS AND

) INTERFERENCES
)
)
LEE E. BARRETT)
Administrative Patent Judge)

Frederick S. Burkhart
Van Dyke, Gardner, Linn & Burkhart
2851 Charlevoix Drive, SE
Suite 207
Grand Rapids, MI 49546